Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claims 1 - 5 (Cancelled):

Claims 6 - 13 (Cancelled):

Claims 14 - 28 (Cancelled):

Claim 29. (New): A compression engine comprising:

an input port for receiving input data vectors;

an output port for providing codevectors and an index map;

volatile memory in data communication with the input port and the output port for storing the input data vectors, the codevectors and the index map;

a codevector trainer in data communication with the volatile memory for training codevectors in dependence upon the input data vectors using one of along vector components codevector training or across vector components codevector training; and,

a controller in communication with the codevector trainer and the volatile memory for executing in cooperation with the codevector trainer one of Successive Approximation Multi-stage Vector Quantization (SAMVQ) or Hierarchical Self-Organizing Cluster Vector Quantization (HSOCVQ) data compression in order to produce compressed data comprising the codevectors and the index map, said controller executing said SAMVQ or HSOCVQ data compression in cooperation with the codevector trainer

wherein said codevector trainer is selected from either a matrix architecture codevector trainer training along vector components or a parallel architecture codevector trainer training across vector components, said matrix architecture codevector trainer comprising :

an input data vector memory for storing a plurality of input data vectors therein;

a codevector memory for storing j codevectors therein;

i*j vector distance calculating units forming an array of i rows and j columns, the i*j vector distance calculating units being in data communication with the input memory and the codevector memory, the i*j vector distance calculating units being for performing the steps of:

a) receiving one component of each of i input data vectors from the input data vector memory and one component of each of the j codevectors from the codevector memory, the components corresponding to a same dimension within the vectors;

b) determining i*j vector component distances corresponding to i*j combinations of the i input data vectors and the j codevectors;

c) summing the successive i*j vector component distances such that each of the successive i*j vector component distances is added in a respective vector distance calculating unit of the i*j vector distance calculating units in order to determine i*j scalar vector distances; and,

d) repeating the steps a), b) and c) for successive components;

e) accumulating the i input data vectors in j partitions, the j partitions corresponding to the j codevectors, the i input data vectors being accumulated in dependence upon their assignment to one of the j partitions; and,

a control unit in data communication with the input data vector memory, the codevector memory, and the i*j vector distance calculating units for performing the steps of:

controlling provision of the components of the input data vectors and the codevectors to the respective vector distance calculating units;

determining a best match codevector for each of the i input data vectors in dependence upon the scalar vector distances;

assigning each of the i input data vectors to one of the j partitions in dependence upon the best match codevector; and,

updating the codevectors in dependence upon the accumulated input data vectors associated with j partitions

and wherein said parallel architecture codevector trainer comprises :

an input data vector memory for storing a plurality of input data vectors therein;

a codevector memory for storing N codevectors therein;

a vector distance calculator for determining a vectored form of a distance between an input data vector and a codevector provided thereto;

a scalar distance block for receiving the vectored form of the distance and for summing vector components of the distance in order to provide a scalar distance;

a minimal distance selector block for determining a minimum scalar distance between an input vector and the N codevectors;

N vector accumulators, each of the N vector accumulators for receiving the input data vectors associated with one of the N codevectors based upon the minimum scalar distance and for accumulating the received input data vectors;

a codebook update process block for updating the N codevectors based upon the accumulated input data vectors; and,

an exit criteria block for determining a termination of the codevector training process based upon the minimum scalar distances and a predetermined threshold.

30. (New): A compression engine as defined in claim 29 further comprising a programming bus connected to a programming port, the codevector trainer, the controller, and the volatile memory for transmitting a control signal.

31. (New): A compression engine as defined in claim 30 wherein the data compression is performed within the compression engine without external communication during the compression process.

32. (New): A compression engine as defined in claim 31 comprising a clock domain decoupled from a clock domain of a system environment with which it is in communication.

33. (New): A compression engine as defined in claim 32 wherein the hardware components of the compression engine are accommodated within a single IC.

34. (New): A compression engine as defined in claim 33 wherein the IC is a FPGA.

35. (New): A compression engine as defined in claim 29 wherein, for said matrix architecture codevector trainer, i*j vector component distances at a vector dimension are determined per clock cycle.

36. (New): A compression engine as defined in claim 35 wherein the input data vectors comprise spectral vectors.

37. (New): A compression engine as defined in claim 29 wherein said compression engine is one of a plurality of compression engines in a compressor, said compressor comprising :

said plurality of compression engines for simultaneously compressing a plurality of data subsets of a set of input data vectors and providing compressed data thereof using one of SAMVQ or HSOCVQ data compression;

a compressorn input port for receiving the set of input data vectors;

an compressor output port for providing the compressed data;

a network switch in data communication with the compressor input port, the compressor output port, and the plurality of compression engines, the network switch being for performing the steps of:

partitioning the set of input data vectors into the plurality of data subsets;

providing each of the plurality of data subsets to one of the plurality of compression engines; and,

transmitting the compressed data.

38. (New): A compression engine as defined in claim 37 wherein said compressor further comprises a programming bus connected to a programming port, the network switch, and the plurality of compression engines for transmitting a control signal.

39. (New): A compression engine as defined in claim 37 wherein said compressor further comprises an input buffer memory in data communication with the network switch.

40. (New): A compression engine as defined in claim 37 wherein in said compressor hardware components of each of the plurality of compression engines and the network switch are accommodated each on a single IC connected to a PCB board.

41. (New): A compression engine as defined in claim 37 wherein said compressor further comprises a card to card connector for providing data and programming communication to a second PCB board.

42. (New): A compression engine as defined in claim 37 wherein each of the plurality of compression engines in said compressor comprises an independent clock domain.